

IN THE UNITED STATES DISTRICT COURT  
FOR THE EASTERN DISTRICT OF TEXAS  
LUFKIN DIVISION

COMPUTER ACCELERATION  
CORPORATION,

*Plaintiff,*

V.

MICROSOFT CORPORATION

Defendant

§ § § § § § § § § §

Case Nos. 9:06-CV-140

**MEMORANDUM OPINION AND ORDER CONSTRUING CLAIM TERMS OF**  
**UNITED STATES PATENT NO. 5,933,630**

Plaintiff Computer Acceleration Corporation (“CAC”) filed suit against Defendant Microsoft Corporation (“Microsoft”) claiming infringement of United States Patent No. 5,933,630 (“the ‘630 patent”). The court conducted a *Markman* hearing to assist the court in interpreting the meaning of the claim terms in dispute. Having carefully considered the patent, the prosecution history, the parties’ briefs, and the arguments of counsel, the court now makes the following findings and construes the disputed claim terms.<sup>1</sup>

## I. Claim Construction Standard of Review

Claim construction is a matter of law. *Markman v. Westview Instruments, Inc.*, 517 U.S. 370, 116 S. Ct. 1384 (1996) (“*Markman II*”). “The duty of the trial judge is to determine the meaning of

<sup>1</sup>This Order governs in the event of any conflict between the Order and the Court’s preliminary analysis at the hearing. The record including the answers of the parties’ counsel and experts to questions from the court, may be helpful in understanding the parties’ arguments and the reasons for the conclusions set out herein.

the claims at issue, and to instruct the jury accordingly.” *Exxon Chem. Patents, Inc. v. Lubrizol Corp.*, 64 F.3d 1553, 1555 (Fed. Cir. 1995) (citations omitted).

“‘[T]he claims of the patent define the invention to which the patentee is entitled the right to exclude.’” *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312 (Fed. Cir. 2005) (*en banc*) (citation omitted). “Because the patentee is required to ‘define precisely what his invention is,’ it is ‘unjust to the public, as well as an evasion of the law, to construe it in a manner different from the plain import of its terms.’” *Phillips*, 415 F.3d at 1312 (quoting *White v. Dunbar*, 119 U.S. 47, 52 (1886)).

The words of a claim are generally given their ordinary and customary meaning. *Phillips* 415 F.3d at 1312. The “ordinary and customary meaning of a claim term is the meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention.”<sup>2</sup> *Id.* at 1313. Analyzing “how a person of ordinary skill in the art understands a claim term” is the starting point of a proper claim construction. *Id.*

A “person of ordinary skill in the art is deemed to read the claim term not only in context of the particular claim in which the disputed term appears, but in the context of the entire patent, including the specification.” *Phillips*, 415 F.3d at 1313. Where a claim term has a particular meaning in the field of art, the court must examine those sources available to the public to show what a person

---

<sup>2</sup> Based on the patent and its cited references, the tutorials, and the representations of the parties at the hearing, the court finds that “one of ordinary skill in the art” in this case is someone with the equivalent of a “four-year” degree from an accredited institution (usually denoted in this country as a B.S. degree) in a field such as mathematics, computer science, electrical engineering or computer engineering (with a concentration of courses in programming, how operating systems work, and the development and use of hardware and software) and between three and five years of programming experience, with two of those years involving operating systems. Extensive experience and technical training might substitute for educational requirements, while advanced degrees might substitute for some of the experience.

skilled in the art would have understood disputed claim language to mean. *Id.* at 1414. Those sources “include ‘words of the claims themselves, the remainder of the specification, the prosecution history, and extrinsic evidence concerning relevant scientific principles, the meaning of technical terms, and the state of the art.’” *Id.* (citation omitted).

“[T]he ordinary meaning of claim language as understood by a person of skill in the art may be readily apparent even to lay judges, and claim construction in such cases involves little more than the application of the widely accepted meaning of commonly understood words.” *Phillips*, 415 F.3d at 1314. In these instances, a general purpose dictionary may be helpful. *Id.*

However, the Court emphasized the importance of the specification. “[T]he specification ‘is always highly relevant to the claim construction analysis. Usually it is dispositive; it is the single best guide to the meaning of a disputed term.’” *Phillips*, 415 F.3d at 1315 (quoting *Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1582 (Fed. Cir. 1996)). A court is authorized to review extrinsic evidence, such as dictionaries, inventor testimony, and learned treatises. *Phillips*, 415 F.3d at 1317. But their use should be limited to edification purposes. *Id.* at 1319.

The intrinsic evidence, that is, the patent specification, and, if in evidence, the prosecution history, may clarify whether the patentee clearly intended a meaning different from the ordinary meaning, or clearly disavowed the ordinary meaning in favor of some special meaning. *See Markman v. Westview Instruments, Inc.*, 52 F.3d 967, 979-80 (Fed. Cir. 1995). Claim terms take on their ordinary and accustomed meanings unless the patentee demonstrated “clear intent” to deviate from the ordinary and accustomed meaning of a claim term by redefining the term in the patent specification. *Johnson Worldwide Assoc., Inc. v. Zebco Corp.*, 175 F.3d 985, 990 (Fed. Cir. 1999).

The “‘ordinary meaning’ of a claim term is its meaning to the ordinary artisan after reading

the entire patent.” *Phillips*, 415 F.3d at 1321. However, the patentee may deviate from the plain and ordinary meaning by characterizing the invention in the prosecution history using words or expressions of manifest exclusion or restriction, representing a “clear disavowal” of claim scope. *Teleflex, Inc. v. Ficosa N. Am. Corp.*, 299 F.3d 1313, 1327 (Fed. Cir. 2002). It is clear that if the patentee clearly intended to be its own lexicographer, the “inventor’s lexicography governs.” *Phillips*, 415 F.3d at 1316.

## **II. Patent Background and Technology**

The ‘630 patent relates to methods and apparatus for optimizing access to a computer program storage device during program start-up, and more particularly to a method for reducing the time to launch a computer program. This is accomplished by using a software routine that detects whether a computer program is being launched. If there is no final log file for the program being launched, then two steps are required. First, an initial log file is created and populated with log entries by software, which: 1) detects each access made to the secondary storage device during the launch; and 2) appends to the log file a log entry reflecting information about each access to the secondary storage device. Second, the initial log file is altered to eliminate any redundant entries in the log files.

The altered log files are sorted to optimize access time to the secondary storage device. During subsequent launches of the computer program, all of the memory blocks specified in the log entries in the final log file are copied to a RAM cache so that each secondary storage device access generated during the subsequent launch results in these accesses being satisfied by the RAM cache rather than the secondary storage device. In almost all modern applications, the computer processor has faster access to the RAM cache than it does to the secondary storage device so launch time is accelerated.

## II. Claim Construction

Claim 1, a method claim, is set out below with the disputed terms in bold, and the agreed terms in italics:

A method for reducing *launch time* for a computer program, wherein the computer program comprises a plurality of files stored on a *secondary storage device*, each one of the plurality of files stored in a plurality of *physical addresses* encompassing at least a portion of a *physical address block* of the *secondary storage device*, a *physical address block* being a minimum unit of *physical addresses* on the *secondary storage device* copied to a *primary storage device*, the method comprising the steps of:

detecting that a computer is executing a *launch sequence* for initializing the computer program, wherein during the *launch sequence*, contents of *physical address blocks* of the *secondary storage device* at which portions of the computer program are stored are copied to the *primary storage device*;

**logging accesses** to the *secondary storage device* during the computer program *launch sequence*, wherein a **log entry** is generated in sequence for each access to a *physical address block* at which a portion of the computer program is stored, and wherein the sequence of log entries corresponds to an access order of portions of the computer program;

detecting that the *launch sequence* is complete;

**altering the log file to eliminate duplicate accesses to the same *physical address block***;

reordering the remaining log entries in the *altered log file* to generate a *final log file*, wherein the remaining log entries are reordered to reduce access time for performing *secondary storage device* accesses;

**detecting a subsequent *launch*** of the computer program;

**before completing the subsequent *launch*, processing the final log file to move the *memory blocks* specified in the final log file into a RAM cache** so that during recommencement of the subsequent *launch* data specified in *secondary storage device* access requests are present in the RAM cache.

Claim 2 is an apparatus claim, which reads as follows:

An apparatus for reducing *launch time* for a computer program, comprising:

a first processor for executing a *launch sequence* of the computer program;

a RAM cache serving as a *primary storage device* accessed by the first processor;

a *secondary storage device* for permanently storing the computer program, wherein the computer program comprises a plurality of files stored on the *secondary storage device*, each one of the plurality of files stored in a plurality of *physical addresses* encompassing at least a portion of a *physical address block* of the *secondary storage device*, a *physical address block* being a minimum unit of *physical addresses* on the *secondary storage device* copied to a *primary storage device*;

means for detecting that the first processor is executing a *launch sequence* for initializing the computer program, wherein during the *launch sequence*, contents of *physical address blocks* of the *secondary storage device* at which portions of the computer program are stored are copied to the primary storage device;

means for **logging accesses** to the *secondary storage device* during an initial computer program *launch sequence*, wherein a **log entry** is generated in sequence for each access to a *physical address block* at which a portion of the computer program is stored, and wherein the sequence of log entries corresponds to an access order of portions of the computer program;

means for detecting that the *launch sequence* is complete;

means for altering the log file to eliminate duplicate accesses to the same *physical address block*;

means for reordering the remaining log entries in the *altered log file* to generate a *final log file*, wherein the remaining log entries are reordered to reduce access time for performing *secondary storage device* accesses;

means for detecting a subsequent *launch* of the computer program; and

means for processing the *final log file*, before completing the subsequent *launch*, to move the *memory blocks* specified in the *final log file* into a RAM cache so that during recommencement of the subsequent *launch* data specified in *secondary storage device* access requests are present in the RAM cache.

### **Common Technical Terms**

The patent includes certain technical terms used in the field of computer science, which were not disputed, but which are helpful to define, as they have meanings commonly understood by one of ordinary skill in the art. The parties' technical experts agreed that while the way they are used in this patent might imply some slight differences, generally these definitions would be accepted by such an artisan for generic use.

1. "log" means a record of transactions or activities that take place on a computer system.
2. "record" means a set of data items, treated as a unit.
3. "file" means a set of related records, treated as a unit.
4. "log file" means a set of related records, called log entries, which is treated as a unit, and which concern transactions or activities that take place on a computer system.
5. "access" means the process of obtaining data from, or placing data into, a storage device, which can include short term, long term, and remote storage.

**1. "Log Entry"** (. . . wherein a log entry is generated in sequence for each access to a physical address block . . .). Claims 1 and 2.

CAC proposes: "information relating to one or more activities that take place on a computer system recorded in a log." Microsoft suggests "an entry in a log file including a file identifier, the physical address(es) specified by a secondary storage device access, and the time of access."

The parties agreed at the hearing that a log entry is information concerning transactions or activities that take place on a computer, which is placed in a log file. The real question is whether, in the context of this patent, a log entry must, at a minimum, include a physical address. The "physical address" is the actual location of a unit of data on the disk or other storage device. This is

contrasted with the “logical address” which is the address the computer program uses to access memory. It is an identifier for the unit of data, which is independent of the data’s physical location.

Microsoft argues that the claim language mandates that every log entry include a physical address, pointing to the fact that the words “logical address” are not used in either claim. The inclusion of “physical memory address(es)” in the log entry is described in the ‘630 patent, col. 5, ll. 24-26. But the sentence starts with “Alternatively.” The specification also describes another embodiment in which the “log entry includes a file identifier, the logical address(es) specified in the call and the time of access (e.g., system time; index value). ‘630 patent, col. 5, ll. 22-24.

It is true that in one embodiment, “all the log entries are sorted according to physical address to optimize access time . . . .” ‘630 patent, col. 7, ll. 62-63. Again, that sentence begins with “Alternatively.” The same paragraph also states: “Each access resulting in a log entry specifies a file and an address.” ‘630 patent, col. 7, ll. 56-57. The indefinite article “an,” without more, does not connote a single embodiment. *See Scanner Technologies Corp. v. ICOS Vision Systems Corp., N.V.*, 365 F.3d 1299, 1304-1305 (Fed. Cir.2004)(construing “an” in a claim).

Under the patent, there will be log entries made, altered, and reordered. There is no basis for the court to decide, as a matter of claim construction, that the definition of “log entry” must include a physical address. The experts may testify that, for technical reasons, based upon the remaining language of the claim describing “logging accesses to the secondary storage device” at ‘630 patent, col. 8, ll. 48-54, including a physical address in every log entry is the only practical application of the invention. That is a matter for trial, not claim construction.



Disclosure-Dedication Argument

\_\_\_\_\_ Relying upon *Toro Co. v. White Consol. Indus., Inc.*, 383 F.3d 1326, 1331 (Fed. Cir. 2004), and *Johnston & Johnston Assoc. Inc. v. R.E. Serv. Co., Inc.*, 285 F.3d 1046, 1054 (Fed. Cir. 2002), Microsoft argues that the disclosure of logical addresses in the specification, combined with the failure to state them in a claim, results in a donation of that embodiment to the public.

\_\_\_\_\_ The specification must give a full and exact description of claimed invention. *Phillips*, 415 F.3d at 1316. The purpose of the specification is to teach and enable those of skill in the art to make and use the invention and provide the best mode for doing so. *Id.* at 1323. Therefore, it is not unusual that the specification contains more detail than the claim, or describes alternative embodiments. Microsoft admits that this is not a disclaimer or disavowal situation, where, in order to overcome a rejection from the patent office, the patentee changed a claim or expressly stated in a response to the PTO that the claims do not include a piece of prior art. However, like prosecution estoppel, the disclosure-dedication rule limits the application of the doctrine of equivalents. *Toro*, 383 F.3d at 1331. Claim construction is not the time for the court to decide infringement issues.

Neither the language of the claim nor the specification indicates any special use in this claim term of any of the common technical terms set out above. There is no basis to import a limitation from the specification to this claim term. Whether CAC can show how the patented method can work with log entries that do not have a physical address, or whether Microsoft can show that an infringement by equivalents argument is barred, are questions left to a later time. The court defines this term as follows:

**“Log Entry”** means: A set of data items, treated as a unit, which concern transactions or activities that take place on a computer and which is placed into a set of related records concerning such transactions or activities.

**2. “Log file”** (. . . altering the log file to eliminate duplicate accesses . . .). Claims 1 and 2.

CAC states that “log file” means “a file that includes information relating to one or more activities that take place on a computer system.” This is just the generic definition set out above. But the log file has a particular meaning in this case - it is what is developed by the patented method to increase launch speed. However, it is not specifically defined in the claim or specification. The first time the term is used is in the fourth step of the claimed method, when it is being altered. It is appropriate to define what the log file is, to make sense of what is being altered.

Microsoft proposes “a file, created during an initial launch time of a computer program, that contains all log entries created during the launch time of the computer program, in the order created.” The problem with stating the precise contents of the “log file” in this method, is that the term is used several times in the claim, with different modifiers at different steps of the method, including:<sup>3</sup>

- a. Fourth step - “altering the log file,” ‘630 patent, col. 8, l. 56;
- b. Fifth step - “entries in the altered log file,” ‘630 patent, col. 8, l. 58; and
- c. Seventh step - “processing the final log file,” ‘630 patent, col. 8, ll. 63-64.

As noted above, under “Common Technical Terms,” one of ordinary skill in the art would know that a computer log file is made up of records called “log entries.” The third step of Claim 1 describes the initial creation of the log file, stating that “a log entry is generated in sequence for each access to a physical address block . . . .” ‘630 patent, col. 8, ll. 49-51.

CAC at first argued that the log file did not have to be created during the initial launch. However, the flow chart at Figure 3 shows that when the launch of a program is directed, the first

---

<sup>3</sup>Each step in Claim 1, a method claim, is mirrored by corresponding means clause in Claim 2, an apparatus claim. For convenience references will be only to the steps in Claim 1.

question asked is whether a log file exists. If not, the log file is created. Fig. 3, Blocks 72, 74, & 76. CAC tried to argue that this was just an embodiment. But, neither its attorneys nor its experts could give an example of a single embodiment described in the patent, or logically derived from the claims, in which the log file was not created sometime during the initial launch.

It seems that CAC's argument is really an attempt to anticipate a non-infringement argument concerning a method in which something is done to the log file between the time of its creation (Claim 1, Step 2) and its alteration (Claim 1, Step 4). That issue will have to be reserved for a later time. At this point, the court is merely defining what a log file is in this patent, not what may be done to it after it is created.

Microsoft urged a construction by which the log file would contain all log entries in the order created. But nothing in either Claim 1 or 2 indicates that when the log entries are first placed in the file (the "logging" of Step 2), they will be kept in any order. Likewise the specification has no such provision.<sup>4</sup> It is true that at Step 5, a "final log file" is generated from the "altered log file," by reordering log entries to speed up a subsequent launch. '630 patent, col. 8, ll. 58-61. But that is described in the claim language, and is not part of a proper definition of "log file."

One issue was not addressed in the briefs, but is apparent from the flow chart at Figure 3. The first question asked in the flow chart is: does the log file exist? Figure 3, item 74. There is no reason to limit the creation of the log file to the very first time the program is ever used, as might be implied from the term "initial launch," which was proposed by Microsoft. The log file could have been

---

<sup>4</sup>In describing the construction and operation of computers, it could be misleading in the context of this patent, especially to those not skilled in the art, such as jurors, to talk of information being "stored in order." The data structure into which bits and bytes of information are placed may be accessed in a particular order, but that is not the same as physically storing them in order, like packages stacked in a box.

deleted accidentally, or because of a cleanup of material that had not been used for a long period of time. In such a case, the log file would be created even though it was not the “initial” launch of the program. It is more accurate to describe a log file as one which is created if it does not already exist.

Based upon the foregoing, the representations of the parties at the claim construction hearing, the agreed definitions of “launch” and “launch sequence,” and the way one of ordinary skill would generically understand “log entry” and “log file,” the court defines this term as follows:

**“Log file”** means: the set of log entries, which, if the log file for a program does not already exist on the computer system, are, upon launch of that program, generated in sequence for each access during the launch sequence to a physical address block at which a portion of the computer program is stored.

**3. “Logging accesses to”** the secondary storage device . . . (Second step of Claim 1 and second means element of Claim 2).

Although the parties initially asked the court to construe “logging access,” the parties agreed at the *Markman* hearing that the phrase, in context, is actually “logging accesses to” the secondary storage device. The parties further agreed that “logging accesses to . . .” can be replaced with the phrase, “placing in a log file, information about the obtaining of data from . . .” This comports with the specification and the meaning of the words commonly understood by those of ordinary skill in the art. The court defines this term as follows:

**“Logging accesses to . . .”** means: Placing in a log file, information about the obtaining of data from . . .

**4. “Detecting a subsequent launch.”** (Sixth step of Claim 1 and sixth means element of Claim 2).

The specification describes one way a subsequent launch may be detected. ‘630 patent, col. 4, ll. 54-63. Grammatically, the way the claims are written, this term refers to a launch of a program which occurs after the one which resulted in the creation of the final log file. *See* ‘630 patent, col. 8, l. 5 (stating the file is closed as step 98 of Fig. 5). The same step is shown in the flow chart at Figure 3. If a launch is detected (Fig. 3, item 72) and a log file exists (Fig. 3, item 74), then before the launch sequence executes, the computer program takes the steps for a more rapid launch (Fig. 6, item 102).

CAC suggested “detecting a launch after the log file is created.” Microsoft proposed “identifying a launch of a computer program after the final log file for that computer program has been created, stored, and associated with that computer program.” After some discussion with the court, the parties agreed on the following definition which the court finds is supported by the claims and specification and comports with the usage of words as understood by one of ordinary skill in the art:

**“Detecting a subsequent launch”** means: Detecting a launch of a computer program after the final log file for that computer program has been generated and stored in RAM, RAM cache, secondary storage, or some other storage.

**5. “Before completing the subsequent launch, processing the final log file to move the memory blocks specified in the final log file into a RAM cache”** (Seventh step of Claim 1 and seventh means element of Claim 2).

The parties have agreed to the court’s construction of “subsequent launch” as “a launch of a computer program after the final log file for that computer program has been generated and stored in RAM, RAM cache, secondary storage, or some other storage.” The “final log file” is “the log file

processed so that log entries or log entry portions that would cause duplicate accesses to the same physical address block have been eliminated.” The parties have also agreed that “memory blocks” means the smallest number of physical addresses that can be read from or written to the secondary storage device. The only dispute is: what does “processing . . . to move” mean in the context of these other agreed terms?

CAC proposed “prior to completing the subsequent launch, using a portion of the final log file to move memory blocks into all or a portion of the random access memory.” The court pointed out that a log file doesn’t “move” memory blocks or data. The processor acts on the log file to manipulate data. CAC then proposed “prior to completing the subsequent launch using the final log file to move memory blocks specified in the final log file into the RAM cache.”

This formulation is still somewhat confusing. CAC, of course, admits that the log file doesn’t move or manipulate data. One skilled in the art would probably intuit that the definition was inelegantly phrased and had to imply that the processor acted on the log file. But the court is defining terms to help a jury understand what the terms would mean to one of ordinary skill. The jurors are not likely going to be skilled artisans in the relevant art. So the definitions should not depend upon the jurors being able to imply meaning from some pre-existing knowledge of the way in which computer processing units manipulate data.

Microsoft states that the term means, “during the subsequent launch, reading, in order, each log entry in the final log file and copying, in the same order, the memory blocks specified in each log entry in the final log file from the secondary storage device to the RAM cache.” The action described in this claim term is “processing the final log file to move the memory blocks . . . .” Nothing in the claim term requires movement in a particular order from secondary storage to RAM cache.

The alleged increased speed of the patented method comes from the fact that the program launch is interrupted and the memory blocks specified in the log file are moved into RAM cache. Then, as the launch of the program recommences, requests for information go to the RAM cache, where the information is now stored in optimal order (in method step 5), and from which redundant accesses have been eliminated (in method step 4). *See* ‘630 patent, col. 8, ll. 19-26.

It is perhaps true that the preferred embodiment, i.e. the method with the most speed, might read and copy the log entries from the secondary storage device to RAM cache in order. But that is not required by the claim language. For one thing, the agreed definition of “subsequent launch” includes a launch after the final log is stored in RAM cache. There is nothing in the claims or specification that requires, or even hints, that some of the memory blocks required for program launch may not also be, at that time, still in RAM cache. In that situation, there would be no need to copy anything in order from a secondary storage device to RAM cache. One example would be the relaunch of a program before the computer has emptied the cache from the previous launch.

The court defines this term as follows:

**“Before completing the subsequent launch, processing the final log file to move the memory blocks specified in the final log file into a RAM cache” means:** during the subsequent launch, reading each log entry in the final log file and copying those memory blocs specified in log entries that are not already in the RAM cache, from the secondary storage device to the RAM cache.

### **MEANS-PLUS-FUNCTION CLAUSES**

The remaining terms the parties ask the court to construe involve means-plus-function clauses under 35 U.S.C. § 112(6). Where a claim includes the word “means,” a presumption is

invoked that § 112(6) applies. *See Harris Corp. v. Ericsson Inc.*, 417 F.3d 1241, 1248 (Fed. Cir. 2005). This presumption may be rebutted if the claim recites “sufficient structure for performing the claimed function . . . .” *Id.*

Determining the claimed function and the corresponding structure of means-plus-function clauses are matters of claim construction, so it is appropriate to deal with these issues at the *Markman* stage. *WMS Gaming Inc., v. Int’l Game Tech.*, 184 F.3d 1339 (Fed. Cir. 1999). Claim construction of a means-plus-function limitation involves two steps. *See Medical Instrumentation and Diagnostics v. Elekta AB*, 344 F.3d 1205, 1210 (Fed. Cir. 2003). The court must first identify the particular claimed function, and then look to the specification and identify the corresponding structure for that function. *Id.* “Under this second step, ‘structure disclosed in the specification is corresponding structure only if the specification or prosecution history clearly links or associates that structure to the function recited in the claim.’” *Id.* (citations omitted). “While corresponding structure need not include all things necessary to enable the claimed invention to work, it must include all structure that actually performs the recited function.” *Default Proof Credit Card System, Inc. v. Home Depot U.S.A., Inc.*, 412 F.3d 1291, 1298 (Fed. Cir. 2005).

Since the functions at issue are computer implemented, the patent must disclose an algorithm to be performed by the computer to accomplish the recited functions. *WMS Gaming Inc.*, 184 F.3d at 1349. This does not mean that the patentee must disclose specific source code for the computer. The term “algorithm” is not limited to a formula of mathematical symbols. For example, the steps, formula, or procedures to be performed by the computer might be expressed textually, or shown in a flow chart. *See Application of Freeman*, 573 F.2d 1237, 1245 - 46 (C.C.P.A. 1978) and cases cited therein. But, in whatever format, the structure, in this case a computer which executes an algorithm, must be sufficiently disclosed so that one of ordinary skill in the art can determine the limitations



on what is claimed. *See Budde v. Harley-Davidson, Inc.*, 250 F.3d 1369, 1381-82 (Fed. Cir. 2001); *see also In re Dossel*, 115 F.3d 942, 946-47 (Fed. Cir. 1997).

**1. “Means for detecting that the first processor is executing a launch sequence for initializing the computer program, wherein during the launch sequence, contents of physical address blocks of the secondary storage device at which portions of the computer program are stored are copied to the primary storage device.”** (First means element of Claim 2).

The means term is written plainly using common words with meanings to which the parties agree. Not surprisingly, the parties agree, and the court finds, that the function is “detecting that the first processor is executing a launch sequence for initializing the computer program.”

CAC proposes for its first alternative that the corresponding structure is “software that monitors computer activity to determine when a computer program is being launched.” This tells the reader virtually nothing about the limits of this claim language. It would include any and all software which could accomplish the function in any way.

As noted, the requirement to disclose structure when the function is implemented by a computer does not require the patentee to “disclose exactly what mathematical algorithm will be used . . . .” *In re Dossel*, 115 F.3d at 946. However, that Court stated that the specification described a device that receives data from two sources and then “computes, from the received data, the current distribution by mathematical operations including a matrix inversion or pseudo inversion, and then outputs the result to a display.” *Id.* at 946. The specification also said “‘known algorithm’ could be used to solve the standard equations which are known in the art.” *In re Dossel*, 115 F.3d at 946. While not a precise mathematical formula or flow chart, this description is far more detailed than the bare repetition of the function in the claim term now at issue in the '630 patent.

CAC argues that “computer code” is not a generic term, but rather recites structure that is understood by those of skill in the art to be a type of device for accomplishing the stated functions. *See Affymetrix, Inc. v. Hyseq, Inc.*, 132 F.Supp.2d 1212 (N.D. Cal. 2001). *Affymetrix* can be distinguished because the claims in that case were not stated in “means for” language. *Id.* at 1231. Therefore, there was no presumption that 35 U.S.C. § 112 applied. *Id.* at 1232. The court held that such a claim was not transferred into a means plus function claim simply by the inclusion of the words “computer code.” *Id.* at 1232. Here the claim terms specifically are in means plus function format, and both parties have agreed that 35 U.S.C. § 112(6) applies. *See Wang Lab., Inc. v. America Online, Inc.*, 197 F.3d 1377, 1385 (Fed. Cir. 1999)(citing *WMS Gaming* as good law).

CAC also points to ‘630 patent, col. 2, ll. 3-5 as an alternative embodiment of structure. ‘630 patent, col. 2, ll. 3-5 states, “computer activity is monitored to determine when a computer program is being launched.” CAC argues that this is a sufficient disclosure of a structure that will perform the function because it states *why* the computer activity is being monitored. However, the cited portion of the specification fails to state *how* to monitor the computer activity. Again there would be no limit to the kinds of programs which could be used.

Finally CAC presents, as an alternative, the algorithm stated at ‘630 patent, col. 4, ll. 56-60. Microsoft’s proposal is taken from that part of the specification: “a routine for generating an interrupt each time a file is opened and an interrupt service routine that checks to see if the file is being opened with an ‘execute’ privilege.”

Microsoft’s restatement of the wording in the specification could make it sound as though there were two routines, which is not required by the language of the specification. The court concludes that the structure associated with this means plus function term is as set out in the ‘630 patent, col. 4, ll. 55-60. (“According to . . . with an ‘execute’ privilege”).

**2. “Means for logging accesses to the secondary storage device during an initial computer program launch sequence, wherein a log entry is generated in sequence for each access to a physical address block at which a portion of the computer program is stored, and wherein the sequence of log entries corresponds to an access order of portions of the computer program.”** (Second means element of Claim 2).

The parties agree, and the court finds based on the common meanings of the words and the definitions previously noted, that the function is “logging, in a log file, accesses to the secondary storage device during an initial computer program launch sequence, wherein a log entry is generated in sequence for each access to a physical address block at which a portion of the computer program is stored, and wherein the sequence of log entries corresponds to an access order of portions of the computer program.”

CAC first argues that the corresponding structure is “software that logs accesses to the secondary storage device during launch.” CAC initially stated that Figure 4, in and of itself, identifies an algorithm for “logging accesses.” Figure 4 describes only a flow chart of steps for generating log entries, and fails to disclose what is technically required to perform the steps within the flow chart, namely box 84 and 86. For the reasons stated above, this is not a sufficient disclosure of structure. Both parties’ technical advisors stated that it would require additional research and work to prepare a program to perform the recited function. CAC eventually agreed that ‘630 patent, col. 5, ll. 16-21 more thoroughly describes Figure 4 and sets out the corresponding structure.

Microsoft contends that the corresponding structure is:

An interrupt service routine for creating a log file when a computer program is launched for the first time. After the log file is created, the interrupt service routine then sets a flag to indicate that logging is enabled for such program. A separate routine then monitors all file system activity and each access to the secondary storage device is analyzed to determine to which computer program being launched the particular access pertains. If the access pertains to a computer program being launched for the first time, then a log entry is appended to the appropriate log file. The separate routine sequentially generates log entries in the program’s log file. The routine creates a log entry in the log file each time a computer program accesses the

secondary storage device (including reads, writes, opens, and closes). Each log entry includes a file identifier, the physical address(es) specified by a secondary storage device access, and the time of the access.

Microsoft points to the '630 patent, col. 4, l. 64 - col. 5, l. 28, which among other things describes creating a log file. Microsoft's wording of the specification also includes a separate routine that sequentially generates log entries in the program's log file. The agreed function does not include either creation of a log file or a separate routine for sequential generation of log entries.

The court finds that the corresponding structures to this means plus function term are disclosed in '630 patent, col. 5, ll. 17-31 together with Figure 4. The first embodiment is disclosed in '630 patent, col. 5, ll. 17-24. (Beginning with "referring to FIG 4 routine **82** . . ." and ending with "... index value"). An alternative embodiment describes physical memory addresses corresponding to the logical addresses that are stored in the log entry. '630 patent, col. 5, ll. 24-31. ("Alternatively . . . physical addresses").

**3. "Means for detecting that a launch sequence is complete."** (Third means element of Claim 2).

The parties agree that the function is "detecting that the launch sequence is complete." Based on the definitions of terms previously stated, there can be little dispute that this is what the claim says, and the court adopts this as the function.

CAC initially stated that the structure is "software that determines program launch is complete." This merely restates the function. For the reasons stated above, this is not a sufficient disclosure of structure.

Microsoft urged that the corresponding structure is: "(1) An interrupt service routine that detects that a specific function of an operating system (e.g., 'Get Next Event' for a Macintosh) is called by the computer program when the launch sequence is complete. The interrupt service routine

clears the logging enabled flag for the corresponding computer program. (2) A routine for monitoring access activity to the secondary storage device to determine when activity has ceased for a threshold length of time (e.g., 3 seconds). (3) A routine for monitoring access activity to the secondary storage device to determine when activity has gone below a threshold data throughput rate (e.g., 50 kilobytes per second) for a threshold period of time (e.g., 5 seconds).” This is a restatement of three algorithms identified in the specification. Rather than restate them, or combine them, it is preferable to identify the algorithms as stated in the specification.

The parties agreed, and the court finds, that the corresponding structures are three alternative algorithms disclosed in col. 5, ll 64 - col. 6, ll. 11.<sup>5</sup> The first algorithm describes an “interrupt service routine,” which clears the logging enabled flag for the corresponding application program. ‘630 patent, col. 5, ll. 64 - col. 6, l. 2. (“According to . . . application program”).

The second embodiment is disclosed in ‘630 patent, col. 6, ll. 3-6, which describes how access activity to the secondary storage device is monitored to determine when activity has ceased for a threshold length of time. (“In an . . . seconds”).

The third alternative is stated in ‘630 patent, col. 6, ll. 6-11, which describes an algorithm that monitors insufficient activity to detect when a launch sequence is complete. (“Alternatively . . . to be complete”).

---

<sup>5</sup>The court understands that CAC’s agreement is subject to the court’s ruling that its broad statement of software as function does not meet the requirement of the *WMS Gaming* line of cases that a sufficient algorithm be stated, and that CAC has not waived this point.

**4. “Means for altering the log file to eliminate duplicate accesses to the same physical address block.”** (Fourth means element of Claim 2).

The parties agree that the function is “altering the log file to eliminate duplicate accesses to the same physical address block.” Again, based on the common meanings of the words and the definitions set out previously, this is the plain language of the claim, and will be adopted by the court.

CAC proposed that the structure is “software that eliminates any log entries or log entry portions to redundant memory blocks.” This is just a recitation of function and, under the analysis set out above, does not state a sufficient algorithm.

CAC also identified as structure Figure 5 and ‘630 patent, col. 6, ll. 56-62 which states, “[E]ntries in the log file are tested at step **94** to identify any redundant accesses to portions of the computer program. To determine whether a later entry is an access to a redundant portion of the computer program, the cluster address for the access is identified.” Step 94 is a single block in the flow chart of figure 5. Its title is almost identical to the statement of function agreed to by the parties. It does not tell the skilled artisan how the function is accomplished - or it claims each and every possible way of programming a computer to accomplish the function.

Microsoft suggested that the structure is: “A routine that tests the log entries in the log file to identify any duplicate accesses to portions of the computer program by identifying the cluster address for the secondary storage device access specified in each log entry. The routine compares the cluster addresses to determine if any are the same. If any of the cluster addresses are the same, the routine deletes from the log file all the log entries having duplicate cluster addresses. The routine thus creates an altered log file.” Essentially, this is a restatement of ‘630 patent, col. 7, ll. 15-45.

The parties agree (with CAC's agreement subject to the court's ruling as noted in footnote 5) that the specification discloses as structure the algorithm contained in '630 patent, col. 7, ll. 15-45. ("Following is a . . . (step 94) is complete"). The court identifies this section of the patent as the structure disclosed.

**5. "Means for reordering the remaining log entries in the altered log file to generate a final log file, wherein the remaining log entries are reordered to reduce access time for performing secondary storage device accesses."** (Fifth means element of Claim 2).

The parties agree and, based on the common meanings of the words and the definitions set out previously the court finds, that the function is "reordering the remaining log entries in the altered log file to generate a final log file, wherein the remaining log entries are reordered to reduce access time for performing secondary storage device accesses."

CAC argues that the corresponding structure is "software that sorts the log entries to optimize secondary storage device accesses." For the reasons stated above, this is not a sufficient disclosure of structure.

CAC also identifies five alternate algorithms disclosed in the specification. Two of these algorithms are contained in '630 patent, col. 7, ll. 61-64, and '630 patent, col. 7, l. 64 - col. 8, l. 4. CAC also identifies '630 patent, col. 7, ll 55-60 as an embodiment different than Figure 5 but it merely describes Figure 5.

CAC also points to '630 patent, col. 7, ll. 54-61 as a fifth embodiment because "other criteria" is disclosed in line 61. "Other criteria" is too general to be a sufficient structure because that could include an infinite number of criteria (i.e. ordinal, cardinal, size . . .).

Microsoft states that the corresponding structure is: "A routine that sorts by physical address the log entries in the altered log file. In the routine, the log entries define a queue and rearranged

(sorted) to optimize secondary storage device access time for the accesses specified in the log entries. The sorting is done according to the methods disclosed in U.S. patent application Serial No. 08/656,372, which later became United States Patent No. 5,854,941. The routine thus creates a final log file.” Microsoft argues that the structures identified in ‘630 patent, col. 7, l. 61 - col. 8, l. 4 are the only algorithms that are linked to the function of optimizing secondary storage device accesses. The court is not persuaded. In ‘630 patent, col. 7, l. 54, the specification states, “the modified log file next is sorted at step **96**.” Step 96, as shown in Figure 5, “sort[s] entry in the modified log file to *optimize secondary storage device accesses*.” (emphasis added). Therefore, the embodiment disclosed in ‘630 patent col. 7, l. 54-60 is linked to optimizing secondary storage device accesses.

The court finds that the corresponding structures are three alternative embodiments disclosed in ‘630 patent, col. 7, l. 54 - col. 8, ll. 4 and Figure 5. The first embodiment describes log entries grouped according to the file. Figure 5 and ‘630 patent, col. 7, ll. 54 - 60. (“In one embodiment . . . in the log file”).

The second structure is disclosed in ‘630 patent, col. 7, ll. 61 - 64 and describes log entries sorted according to physical address. (“In an alternative . . . secondary storage device **16**”).

The third alternative is stated in ‘630 patent, col. 7, l. 64 - col. 8, l. 4. (“In one embodiment . . . I/O requests”). This sentence describes log entries sorted by the methods disclosed in U.S. patent application Ser. No. 08/656,372 filed May 31, 1996 for “Estimating Access Time for Hard Drive I/O Requests.”



**6. “Means for detecting a subsequent launch of the computer program.”** (Sixth means element of Claim 2).

The parties agree, and based on the common meanings of the words and the definitions set out previously the court finds, that the function is “detecting a subsequent launch of the computer program.”

CAC initially argued that the structure is “software that monitors computer activity to determine when a computer program is being launched and checks to see if a log file already exists for the computer program being launched.”

Microsoft stated that the structure is: “A routine for generating an interrupt each time a file is opened and an interrupt service routine that checks to see if the file is being opened with an “execute” privilege. The interrupt service routine then checks to see if a final log file already exists for the computer program.”

At the hearing, the parties agreed and the court finds that the specification discloses the structure in col. 4, ll. 54-66. (“The first step . . . being launched”).

**7. “Means for processing the final log file, before completing the subsequent launch, to move the memory blocks specified in the final log file into a RAM cache so that during recommencement of the subsequent launch data specified in secondary storage device access requests are present in the RAM cache.”** (Seventh means element of Claim 2).

CAC argues that the function is “processing the final log file, before completing the subsequent launch, to move the memory blocks specified in the final log file into a RAM cache.”

Microsoft states that the function is “processing the final log file, before completing the subsequent launch, to move the memory blocks specified in the final log file into a RAM cache so that during recommencement of the subsequent launch data specified in secondary storage device access

requests are present in the RAM cache.”

The wording between the parties’ proposals is exactly the same except for the language after “so that” in the Microsoft version. The issue is whether the “so that” clause is merely a statement of the desired result of the limitation in the claim, or whether it is part of the recited function. *See Lockheed Martin Corp. v. Space Sys./Coral Inc.*, 324 F.3d 1308, 1319 (Fed. Cir. 2003)(noting that “a whereby clause that merely states the result of the limitation in the claim adds nothing to the substance of the claim.”)

To support their respective positions, both parties point the court to the same lines of the specification, ‘630 patent, col. 8, ll. 22-24, which states, “As the I/O requests are generated during the remainder of the launch sequence, each request *results* in a hit in the RAM cache.” (emphasis added). This does nothing but point out that the claimed function here, “processing the log file,” results in data requests being present in the RAM cache. Based on this common understanding of the words used and the definitions set out above, the court concludes that the function is, “processing the final log file, before completing the subsequent launch, to move the memory blocks specified in the final log file into a RAM cache.”

CAC argues that the corresponding structure is “software that processes the log file by generating a secondary storage device request for entries in the log file.” For the reasons stated above, this is not a sufficient disclosure of structure.

CAC points to Figure 6 as an alternative algorithm. This flow chart is detailed in ‘630 patent col. 8, ll. 13-21. CAC also identifies ‘630 patent col. 8, ll. 17-21 as an algorithm. This just describes block 106 of Figure 6. It is not, by itself, an alternative algorithm.

Microsoft states that the corresponding structure is “a routine that executes during the

subsequent launch to open the final log file for the computer program being launched. The routine generates a secondary storage device I/O request for each log entry in the final log file in the order the log entries occur within the final log file. The routine causes the memory blocks specified in each log entry to be copied, in the same order, from the secondary storage device to the RAM cache. The routine copies the memory blocks into the RAM cache, so that during the remainder of the launch sequence the memory blocks are present in the RAM cache.” Microsoft cites ‘630 patent, col. 8, ll. 13-21, which is no different than CAC’s proposed structure, except Microsoft adds that log entries are to be copied “in the same order” from the secondary storage device to the RAM cache. Microsoft points out that I/O requests are *generated* in an order, but nothing in the specification states that log entries are to be *copied* in order.

Therefore, the court finds that the corresponding structure to this means plus function claim is described in Figure 6 together with ‘630 patent, col. 8, ll. 13-21. (“The routine . . . RAM code 15”).

#### IV. Conclusion

The jury shall be instructed in accordance with the court’s interpretation of the disputed claim terms in the ‘630 patent.

So **ORDERED** and **SIGNED** this 7 day of **May, 2007**.

A handwritten signature in black ink, appearing to read "Ron Clark", is written over a horizontal line.

Ron Clark, United States District Judge